**Programming Concepts**

**Variables:**

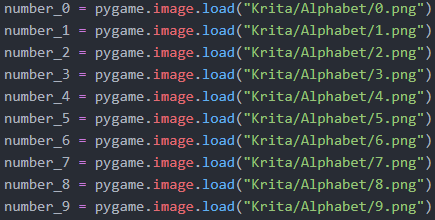
**List**





Used to store the data about the game as it runs through the logic. The different turns can be accessed by using their position in the list. The final value in the list represents what turn the game is on and is always moved along with the code instead of having to use two variables. This allows the current turn to be accessed using code[code[11][0]].

**Dictionary**

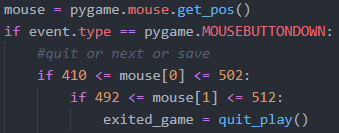






Connects a piece of data to a variable. The variables are predefined and then added into the dictionary with a corresponding piece of data. In this case it is a number and the variable for that number can be accessed using numbers[number].

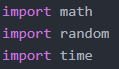
**Array**



The coordinates of the mouse are saved as an array in the format (x,y). They can be accessed in the same way as a list.

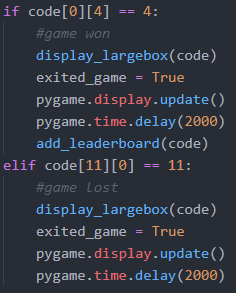
Colours are also stored as an array. The elements can’t be edited which means that the colour will remain the same and not be changed to an unwanted colour.

**Modules**



Import functions and procedures that have been created by other people. Makes writing a program easier as you don’t need to know about every method a computer can perform and instead know how to use a simple command. Also allows for powerful modules such as pygame which give a new set of functions that allow for new ways to interact with a python program.

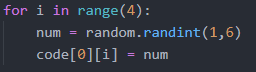
**Selection**



Run a different sequence of commands based on a condition. For example if a game is finished it will run a stop command but if it’s not then it continues the program.

**Iteration:**

**Definite Iteration**



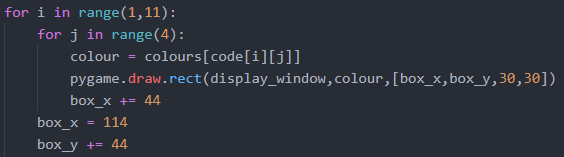
Run the same sequence of commands a set number of times. Removes the need for repetition of the same commands. It can also count up a number which can be used to move along lists or get increasing values to be used in the program.

**Indefinite Iteration**



Run through the same sequence of commands until a condition is met. In this example it is until the game is finished. Until then the same game logic if repeated to run the game.

**Nested Loop**



Used to move around in two dimensions. The first loop can represent the y axis and the other the x axis. In this case it moves across the screen to be able to display coloured squares in a regular pattern.

**Subprograms:**

**Function**

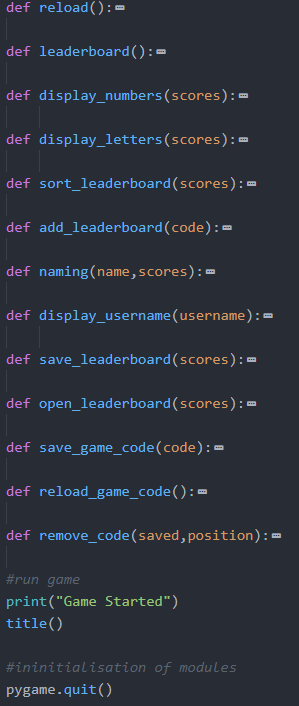
Sub programs that return a value at the end. Things such as opening a file where the data of the file will be returned at the end to allow the rest of the program to use the data. They can be called from many different places removing repetition of the same commands.

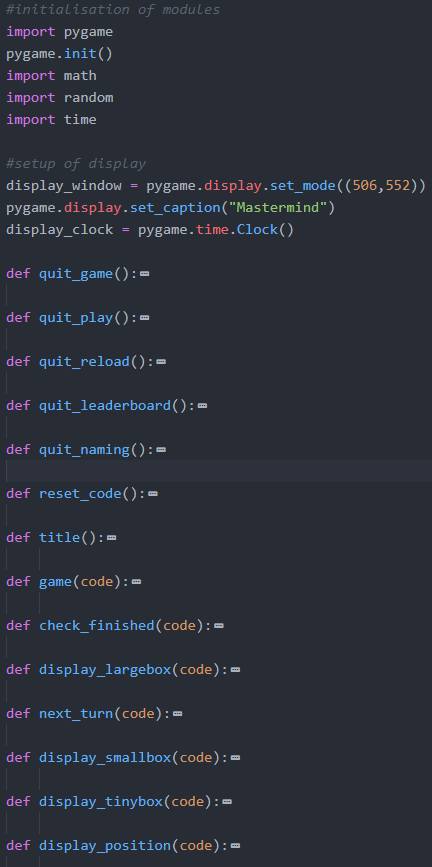
**Procedure**

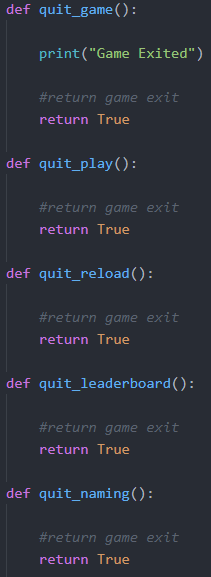
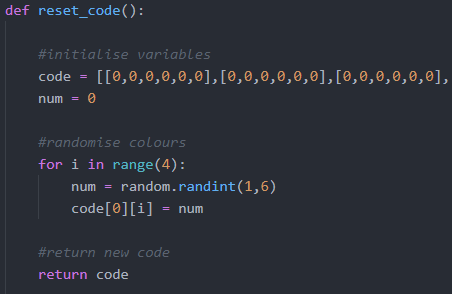
Sub programs only perform a set of commands. Such as displaying something on the screen. It requires a parameter that defined what it should display and then it will display the inputted data before returning to the main program to continue onwards in sequence.

**Examples of running the program**

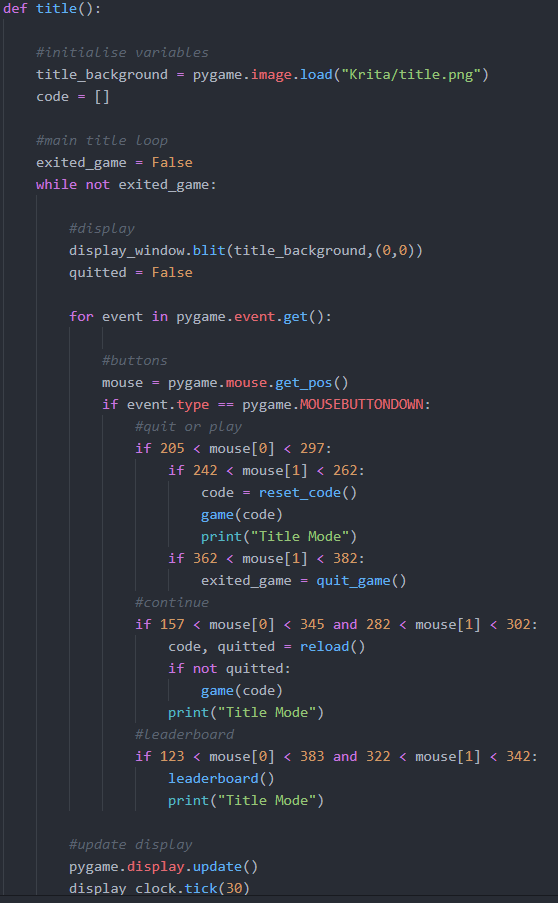
|  |  |
| --- | --- |
| The title page of the program. You can either play, reload a previous game, view the leader board or exit the program. |  |
| The main screen for playing the game. The buttons on the side are to enter a guess for the code, save the game to play later, or to abandon the current game. The top four boxes will show the correct code when the user guesses it correctly or runs out of turns. The coloured boxes on the side allows the user to select a colour to enter their guess.  Pressing the exit button will return the user to the main screen. |  |
| The user clicks on a colour to select it on the right then clock on a box on the left to change it to that colour. Once they have entered a guess, they press the next button to move to the next turn. The turn counter on the side moves down to show what turn the user is on. The small boxes in the middle change to show how many correct colours they have. Red represents a correct colour in the correct position and white represents a correct colour in the wrong position. |  |
| Pressing the save button will add the current game to a text file so that it can be accessed later. This game will be seen later. |  |
| When the game is won it shows what the correct code it at the top to match the users input. It then beings the save game process. |  |
| The user can then enter a 5-letter name to have their score added to. They get two points for every turn away from the last term they were. Pressing back deletes a letter and pressing add will enter the name. All names and scores are saved to a text file. The user is then returned to the main menu. |  |
| In the leader board section of the program it displays the users 5 letter names along with their score. The maximum score is 999. Pressing the exit button will return them to the main menu. |  |
| In the reload screen it shows the previously played games that have not yet been deleted. The user can select exit, delete, or play the game. Also, they can move left and right thought the available games.  Pressing the exit button will return them to the main menu. |  |
| Pressing the delete buttons will get rid of the currently displayed game and move along to the next one. If there are no more games, it returns to the main menu. |  |
| Pressing left or right will move through the list of games and allow the user to select from any previously saved games. |  |
| Pressing the play button will take the currently display game and put in into play mode where the user can continue the game. |  |
| If the user does not get the code after the ten turns it will display the correct answer, then return them to the main menu. |  |

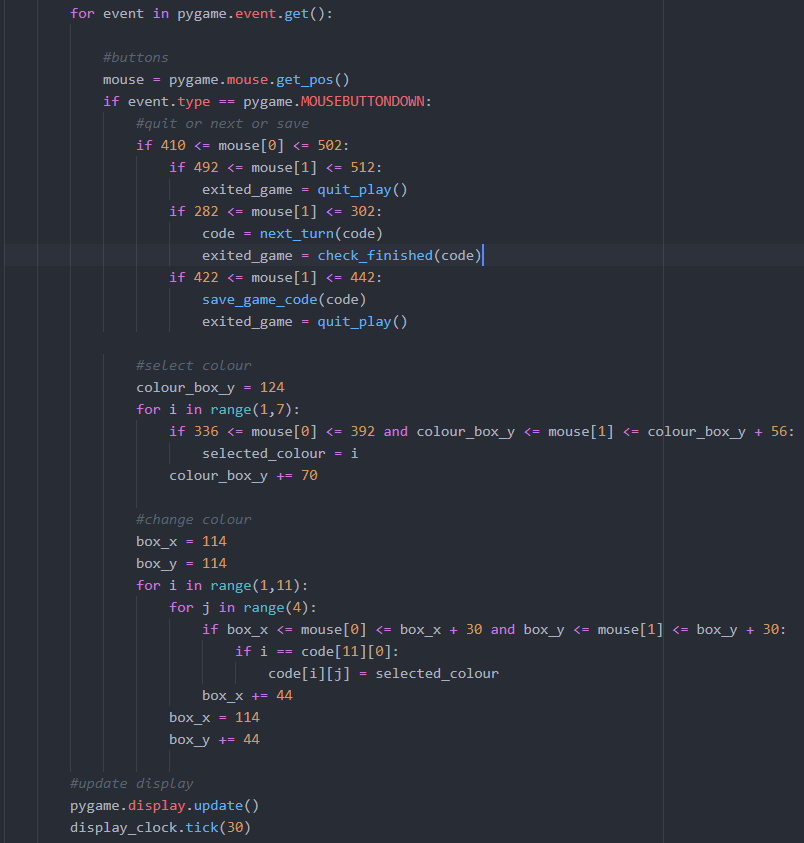
**The code**

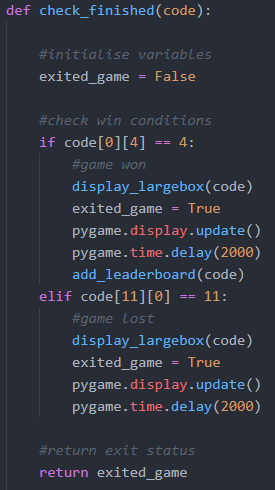


The quit game sub programs are useless

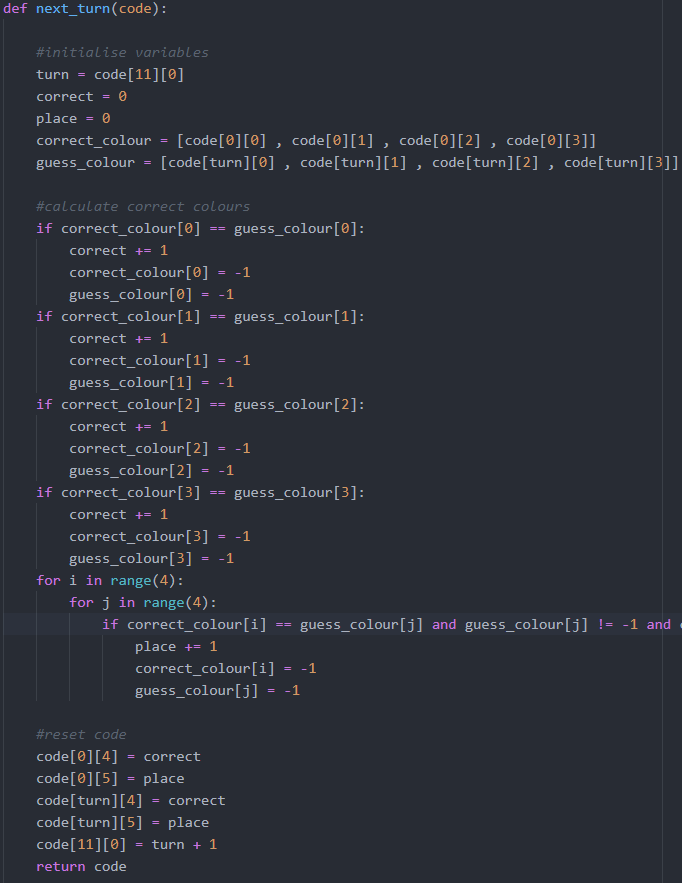
The code list is extended over three screenshots



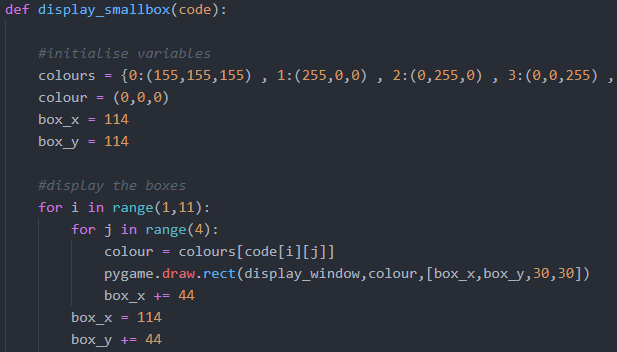


The colours dictionary is extened over two screenshots

The if statement is extended over two screenshots

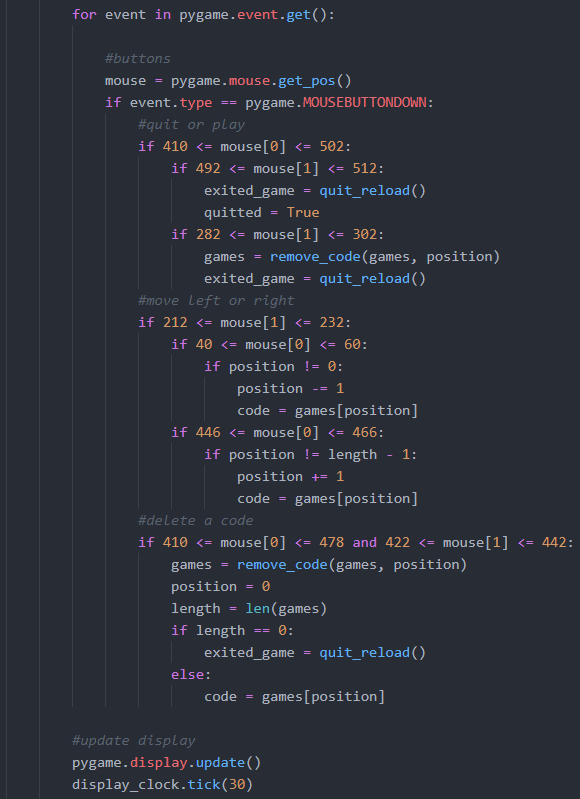
  

The colours dictionary is extended over two screenshots

The positions dictionary is extened over two screenshots

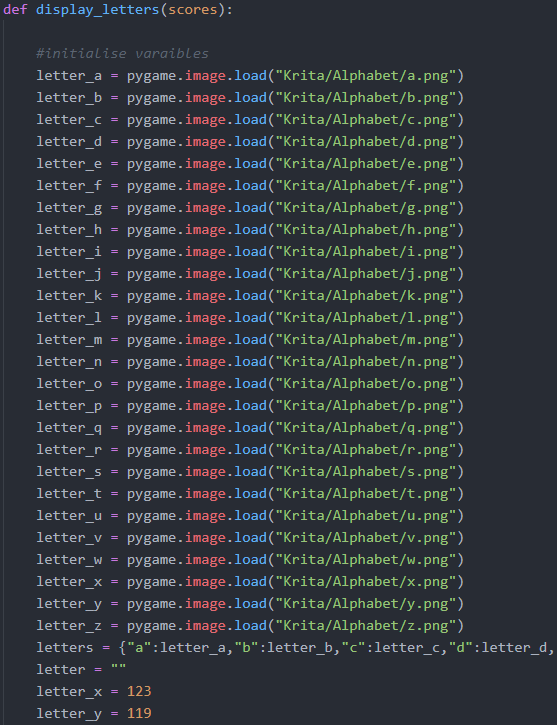
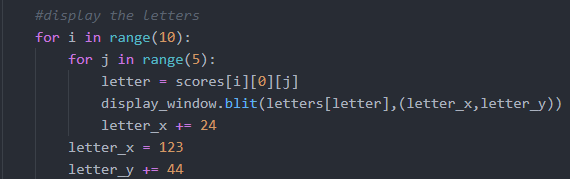




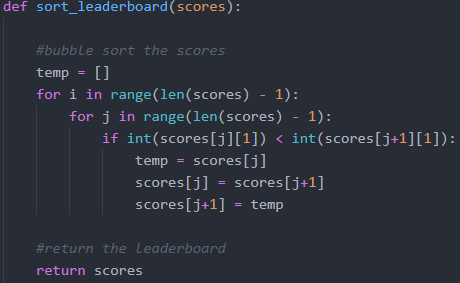
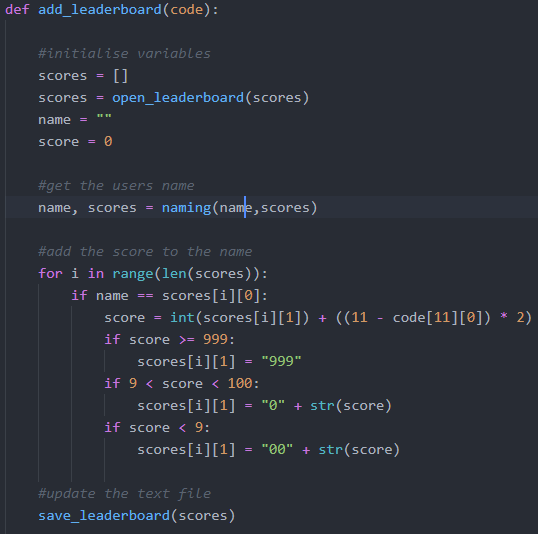
The reload sub program is extended over two screenshots

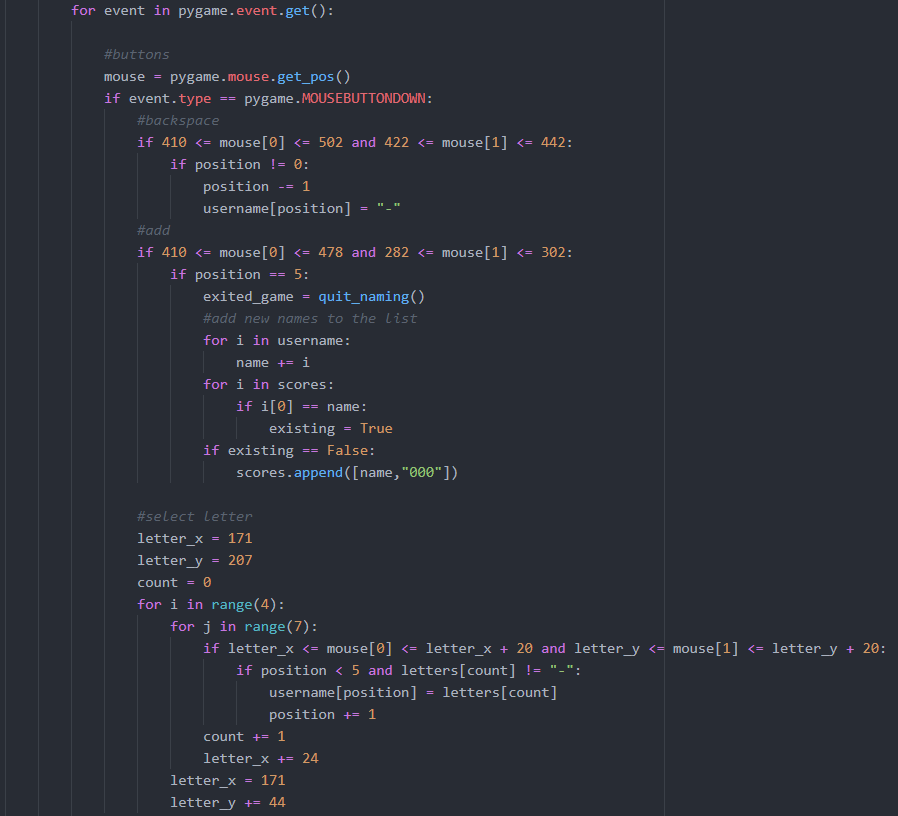
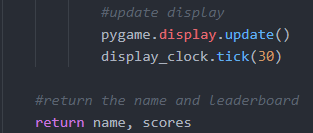
  

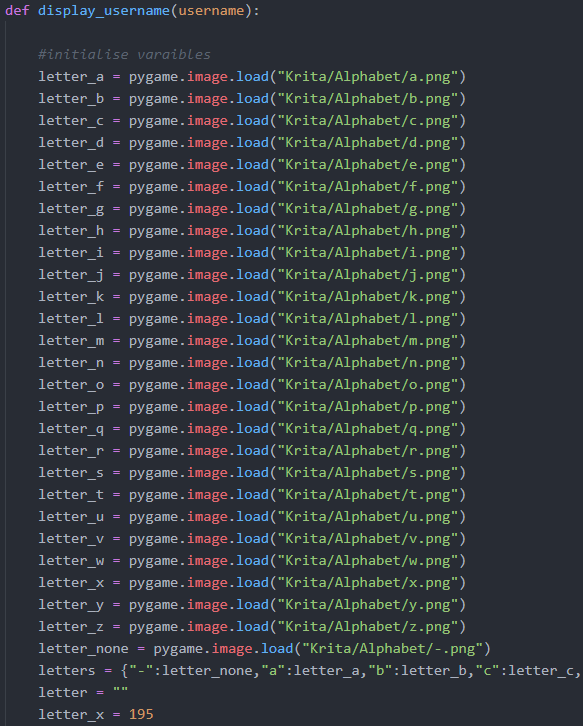
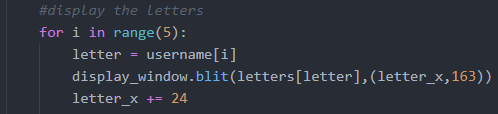
The numbers dictionary is extended over two screenshots

The letters dictionary has the same repeating pattern for all letters of the alphabet

The letters dictionary has the same repeating pattern for all letters of the alphabet

